Appl. No. 10/698,739
Amdt. dated December 1, 2008
Amendment under 37 CFR 1.116 Expedited Procedure
Examining Group 2138

PATENT

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1.   (previously presented)   A method for isolating failed routing resources on a programmable integrated circuit, the method comprising:

receiving a plurality of failed test patterns, wherein a test pattern includes program bits that define how routing resources on the programmable integrated circuit are connected to form a test path, wherein a test pattern is designated as failing when a result from a test path is erroneous, wherein the result of the failed test path is created by applying one or more test values to the failed test path;

identifying a subset of the routing resources, wherein the subset comprises one or more routing resources that respectively occur in the most failed test paths; and

generating new test patterns including program bits that define new test paths for testing a first routing resource of the subset of the routing resources, wherein each of the new test paths includes:

the first routing resource; and

a combination, not included in the other new test paths, of fan-in and fan-out resources that are programmably connectable to the first routing resource,

wherein the new test paths test every combination of fan-in and fan-out resources that are programmably connectable to the first routing resource.

2.   (currently amended)   The method according to claim 1 further comprising:

~~testing~~ running the new test patterns using a test system to isolate routing resources among the subset of the routing resources that caused the erroneous results in the failed test patterns.

3.   (currently amended). The method according to claim 1 wherein generating the new test patterns includes program bits that define new test paths for testing every routing resource of the subset; and

wherein for each routing resource of the subset, the new test paths route through every combination of fan-in resources and fan-out resources that are programmably connectable to that routing resource of the subset.

4.   (original)   The method according to claim 1 wherein generating the new test patterns for the subset of the routing resources further comprises:

generating new test patterns for test paths that route through clock and clear signal routing resources.

5.   (original)   The method according to claim 1 wherein each of the failed test paths and the new test paths connect a control point to an observation point on the programmable integrated circuit.

6.   (original)   The method according to claim 1 wherein the routing resources have more than 1000 times as many routing resources as the subset of routing resources.

7.   (original)   The method according to claim 5 further comprising:

receiving a test log file that indicates the observation points for the failed test paths.

8.   (original)   The method according to claim 1 wherein identifying the subset of the routing resources that occur most frequently in the failed test paths further comprises:

extracting the routing resources that are connected along each of the failed test paths using a connectivity graph.

9.   (previously presented)   A computer program product encoded on a
computer readable medium for isolating failed routing resources on a programmable integrated
circuit, the computer readable medium comprising:

code for receiving a plurality of failed test patterns generating erroneous results
when test values are applied to a set of failed test paths, wherein a test pattern includes program
bits that define how routing resources on the programmable integrated circuit are connected to
form a test path;

code for identifying a subset of the routing resources, wherein the subset
comprises one or more routing resources that respectively occur in the most failed test paths; and

code for generating new test patterns including program bits that define new test
paths for testing a first routing resource of the subset of the routing resources, wherein each of
the new test paths includes:

the first routing resource; and

a combination, not included in the other new test paths, of fan-in and fan-
out resources that are programmably connectable to the first routing resource,

wherein the new test paths test every combination of fan-in and fan-out resources
that are programmably connectable to the first routing resource.

10.   (original)   The computer program product of claim 9 wherein the code
for receiving and identifying further comprises:

code for receiving a test log file that indicates observation points for the failed test
paths.

11.   (original)   The computer program product of claim 9 wherein the code
for generating further comprises:

code for generating new test patterns for test paths that route through clock and
clear signal routing resources.

12.   (original)   The computer program product of claim 9 further
comprising:

code for testing the new test patterns to isolate routing resources among the subset
that caused the erroneous results in the failed test patterns.

13.   (original)   The computer program product of claim 9 wherein the code
for generating further comprises:

code for generating new test patterns that route through every combination of fan-
in resources and fan-out resources that are programmably connectable to each of the subset of
the routing resources.

14.   (original)   The computer program product of claim 9 wherein the
routing resources have more than 10,000 times as many routing resources as the subset of the
routing resources.

15.   (previously presented)   A computer system for isolating failed
routing resources on a programmable integrated circuit, the computer system comprising:

a statistical failure isolation (SFI) tool, wherein the SFI tool:

(a) receives a file including a plurality of failed test patterns that
generated erroneous results when test values were applied to a set of failed test paths, wherein a
test pattern includes program bits that define how routing resources on the programmable
integrated circuit are connected to form a test path;

(b) determines routing resources along each failed test path;

(c) calculates a total number of occurrences of each resource in the failed
test paths received in the file, at least one resource occurring in two failed test paths; and

(d) identifies a subset of the routing resources, wherein the subset
comprises one or more resources having the highest number of occurrences; and

an adaptive failure isolation (AFI) tool that subsequent to completion of (b)-(d) generates new test patterns including program bits that define new test paths for testing the subset of the routing resources.

16.   (original)   The computer system according to claim 15 wherein the SFI tool also receives a test log file that indicates observation points for the failed test paths.

17.   (original)   The computer system according to claim 15 further comprising:

a test system that tests the new test patterns to isolate routing resources among the subset that caused the erroneous results in the failed test patterns.

18.   (original)   The computer system according to claim 15 wherein:

the AFI tool generates new test patterns for new test paths that route through every combination of fan-in resources and fan-out resources that are programmably connectable to each of the subset of the routing resources.

19.   (original)   The computer system according to claim 15 wherein the routing resources have more than 1000 times as many routing resources as the subset of routing resources.

20.   (original)   The computer system according to claim 15 wherein the SFI tool extracts the routing resources that are connected along each of the failed test paths using a connectivity graph.

21.   (previously presented)   The method of claim 1, wherein each of the new test paths is used to determine whether the resources of the subset have actually failed.

22.   (previously presented)   The method of claim 1, wherein the erroneous result of a failed test path is an output value of the failed test path that does not equal an expected value.

Appl. No. 10/698,739
Amdt. dated December 1, 2008
Amendment under 37 CFR 1.116 Expedited Procedure
Examining Group 2117

PATENT

23.   (currently amended)   ~~The method of claim 2,~~ A method for isolating failed routing resources on a programmable integrated circuit, the method comprising:

receiving a plurality of failed test patterns, wherein a test pattern includes program bits that define how routing resources on the programmable integrated circuit are connected to form a test path, wherein a test pattern is designated as failing when a result from a test path is erroneous, wherein the result of the failed test path is created by applying one or more test values to the failed test path;

identifying a subset of the routing resources, wherein the subset comprises one or more routing resources that respectively occur in the most failed test paths; and

generating new test patterns including program bits that define new test paths for testing a first routing resource of the subset of the routing resources,

wherein the testing a new test pattern includes testing ~~tests~~ a clock control point and comprises:

scanning in a first value to a failed resource;

scanning in a second value to a data control point coupled with the failed resource;

scanning out the value stored in the failed resource and comparing that value to the first value;

transmitting a clock signal from the clock control point to the failed resource; and

scanning out the value stored in the failed resource and comparing that value to the second value.

24.   (currently amended)   The method of claim ~~2~~23, wherein the testing a new test pattern includes testing ~~tests~~ a clear control point and comprises:

scanning in a first value into a failed resource;

scanning out the value stored in the failed resource and comparing that value to the first value;

transmitting a clear signal from the clear control point to the failed resource; and

scanning out the value stored in the failed resource and comparing that value to a

clear value.